

Task 7 : Cronjob Hazırlanması : Oluşturulan tablo üzerine Product tipinde yeni bir field açın. Daha sonra yeni bir Cronjob oluşturun ve bu cronjob modeli üzerinde de Product tipinde bir alan olsun. Bu cronjob çalıştırıldığı an, kendi oluşturduğunuz tabloya ait bütün verileri çeksin ve Backoffice'de cronjoba verilen product'ı alarak, tablodaki bütün dataların product alanına set etsin. Cronjobı hazırlarken performanslı ve hızlı olmasını dikkate alın.

The screenshot shows an IDE with a Java project. The active file is `avansascore-items.xml`. The code defines a JPA entity `TrainingProduct` and a cron job `TrainingProductCronJob`.

```
<!-- Task 5 second table -->

<entity code="TrainingProduct" javaClass="com.avansas.core.jalo.TrainingProduct">
  <deployment table="TrainingProduct" typecode="28676"/>
  <attributes>
    <attribute qualifier="productName" type="java.lang.String">
      <persistence type="property" />
      <modifiers optional="false" initial="true"/>
      <description>product name</description>
    </attribute>
    <attribute qualifier="companyName" type="java.lang.String">
      <persistence type="property" />
      <modifiers optional="false" initial="true" />
      <description>company name</description>
    </attribute>
    <attribute qualifier="cronProduct" type="Product">
      <persistence type="property"/>
      <modifiers optional="true" initial="true"/>
      <description>my product table</description>
    </attribute>
  </attributes>
  <indexes>
    <index name="TrainingProductId" unique="true">
      <key attribute="productName" />
      <key attribute="companyName" />
    </index>
  </indexes>
</entity>

<!-- Cronjob and adding an attribute of type product to it -->

<entity code="TrainingProductCronJob" extends="CronJob" autocreate="true" generate="true"
  javaClass="com.avansas.core.jalo.TrainingProductCronJob">
  <attributes>
    <attribute qualifier="cronProduct" type="Product">
```

2- Daha sonra product tipinde bir attribute sahip olana bir CronJob oluşturmak için avansascore-items.xml’de şunu ekliyoruz :

```

2681 <!-- Cronjob and adding an attribute of type product to it -->
2682
2683
2684 <itemtype code="TrainingProductCronJob" extends="CronJob" autocreate="true" generate="true"
2685     jaloclass="com.avansas.core.jalo.TrainingProductCronJob">
2686     <attributes>
2687         <attribute qualifier="cronProduct" type="Product">
2688             <persistence type="property" />
2689             <modifiers optional="true" read="true" write="true"/>
2690             <description>my product in cronjob</description>
2691         </attribute>
2692     </attributes>
2693 </itemtype>
2694
2695
2696
2697
2698

```

TrainingProductCronJob adına cronjob ve cronjob'u extend etmektedir.

ANT CLEAN ALL ve Update yapıyoruz.

```

1 1
2 2
3 3
4 4
5 5
6 6
7 7
8 8
9 9
10 10
11 11
12 12
13 13
14 14
15 15
16 16
17 17
18 18
19 19
20 20
21 21
22 22
23 23
24 24
25 25
26 26
27 27
28 28
29 29
30 30
31 31
32 32
33 33
34 34
35 35
36 36
37 37
38 38
39 39
40 40
41 41
42 42
43 43
44 44
45 45
46 46
47 47
48 48
49 49
50 50
51 51
52 52
53 53
54 54
55 55
56 56
57 57
58 58
59 59
60 60
61 61
62 62
63 63
64 64
65 65
66 66
67 67
68 68
69 69
70 70
71 71
72 72
73 73
74 74
75 75
76 76
77 77
78 78
79 79
80 80
81 81
82 82
83 83
84 84
85 85
86 86
87 87
88 88
89 89
90 90
91 91
92 92
93 93
94 94
95 95
96 96
97 97
98 98
99 99
100 100

```

Modelimiz oluşuyor. Ve tablomuza da eklediğimiz field geldi.

3-Sonra facade/job'da TrainingProductCronJob oluşturuyoruz ve AbstractJobPerformable'yi extend ederek parametre olarak cronJobumuzu veriyoruz.

```

13
14 import javax.annotation.Resource;
15 import java.util.Collections;
16
17 public class TrainingProductCronJob extends AbstractJobPerformable<TrainingProductCronJobModel> {
18
19     @Resource(name = "modelService")
20     private ModelService modelService;
21
22     @Resource(name = "flexibleSearchService")
23     FlexibleSearchService flexibleSearchService;
24
25     @Override
26     public PerformResult perform(TrainingProductCronJobModel cronJob) {
27
28         String query = "SELECT {pn} FROM {trainingProduct} WHERE {cronProduct} IS NULL";
29
30         final FlexibleSearchQuery searchQuery = new FlexibleSearchQuery(query, Collections.singletonMap("cronProduct", cronJob.getCronProduct()));
31
32         final SearchResult<TrainingProductModel> result = flexibleSearchService.search(searchQuery);
33
34         result.getResult().forEach(item -> {
35             item.setCronProduct(cronJob.getCronProduct());
36             modelService.save(item);
37         });
38
39         return new PerformResult(cronJobResult.SUCCESS, cronJobStatus.FINISHED);
40     }
41
42     public FlexibleSearchService getFlexibleSearchService() {
43     }
44 }

```

Bu class'ta AbstractJobPerformable sınıfında gelen perform metodunu override edip içinde bussiness logic'lerimizi yazıyoruz .

Tüm TrainingProduct'ta olan itemsler ve cronProduct'ı null olanları sql ile getiriyoruz daha sonra bu items'lara tek tek cronjob'umuzun product'ımızı get ederek onlara set ediyoruz.En son PerformResult döndürmektedir.Başarıyla çalışırsa SUCCESS ve FINISH gelecektir, bu metodumuz cronjobumuz run edince çalışmaktadır .

CronJob için gereken bean'ı ekliyoruz(avansasfacade-spring.xml'in içinde)

```

4218 <!-- CronJob bean -->
4219 <bean id="TrainingProductCronJob" class="com.avansas.facades.job.TrainingProductCronJob"
4220       parent="abstractJobPerformable"/>
4221
4222
4223
4224

```

#### 4-Impex hazırlanıp basılması :

```

1 INSERT_UPDATE ServiceLayerJob; code[unique = true] ; springId[unique = true]
2 TrainingProductCronJob ; TrainingProductCronJob ; TrainingProductCronJob
3
4 INSERT_UPDATE TrainingProductCronJob; code[unique = true] ; job[code] ; sessionLanguage(isocode)[default=tr]
5 TrainingProductCronJob ; TrainingProductCronJob ; TrainingProductCronJob
6
7
8

```

Cronjobumuz basarıyla backoffice->cronjob'a düřtü. Sonra Cronjob'a bir product ekliyoruz backoffice'den .

Cronjob'u run ediyoruz.

Cronjoba eklediđimiz product bütün tablomuz'daki product'ı null olan items'lara set edildi.

Platform

Monitoring

Maintenance

Console

PayPal Button Settings

Credit Message Configuration

Drop-In Button Customization

FlexibleSearch

Flexible Query

SQL Query

Search result

Execution statistics

History

Commit: OFF

Show10entries

Search:

hjmpTS	createdTS	modifiedTS	TypePkString	OwnerPkString	PK	p_productname	p_companyname	aCLTS	propTS	p_cronproduct
1	2023-03-30 12:00:41.15	2023-03-30 14:06:26.54	8796550070354		8796191387652	moise	samsun	0	0	8823852138497
2	2023-03-30 12:39:39.067	2023-03-30 14:12:15.973	8796550070354		8796224155652	mackbook	apple	0	0	8823852138497

Showing 1 to 2 of 2 entries

Başarıyla çalışmaktadır.